	INFORMATION TECHNOLOGY GUIDELINES	
	Document #:	VCCS-ITS-09-0004
	Contact :	Director, Technology Administrative Services
	Revision:	1.0
	Effective Date:	12/15/2009
	# Pages:	21
INFORMATION TECHNOLOGY ACCESSIBILITY GUIDELINES		

PURPOSE

These guidelines are established to provide clarification, best practices, and procedures for application and web-site development to conform to the VCCS Technology Policy and Standard for accessibility.

SCOPE

This guideline covers:

- Accessibility in Procurement
- What and Why of Accessible Application and Web Design
- Section 508 Explained
- Web Content Accessibility Guidelines
- Tools for developing accessible content
- References

APPLICABILITY

This standard is applicable to the System Office, colleges, departments, auxiliaries, research and administrative entities supporting the programs or services provided by the twenty-three colleges and VCCS.

DEFINITIONS

ICT – Information Communications Technology

Electronic Information Technology – EIT includes information technology and any equipment or interconnected system or subsystem of equipment, which is used in the creation, conversion, or duplication of data or information. The term electronic and information technology includes, but is not limited to: telecommunications products (such as telephones), information kiosks and transaction machines, World Wide Web sites, multimedia, and office equipment such as copiers and fax machines. The term does not include any equipment that contains embedded information technology that is used as an integral part of the product, but the principal function of which is not the acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information. For example, HVAC (heating, ventilation, and air conditions) equipment such as thermostats or temperature control devices, and medical equipment where information technology is integral to its operation, are not information technology.

PDF – Portable Document File – is a file format created by Adobe Systems in 1993 for document exchange. PDF is used for representing two-dimensional documents in a manner independent of the application software, hardware, and operating system.

Section 508 – Section 508 (29 U.S.C.794d) of the Rehabilitation Act 1998 amended.

Section 255 – Section 255 (47 U.S.C. 1193) of the Communications Act of 1996..

GUIDELINES

Introduction	3
Accessibility in Procurement	3
<i>Market Research and Findings</i>	3
<i>Accessible Evaluation Sheet</i>	4
What and Why of Accessible Web Design	4
Making A Web Site Accessible Both For People With Disabilities And For Mobile Devices	4
Section 508	5
Section 508 Described	6
508 (a) <i>Text Equivalent</i>	6
508 (b) <i>Equivalent Multimedia Alternatives</i>	7
508 (c) <i>Color</i>	8
508 (d) <i>Style sheet use</i>	9
508 (e) <i>Server-side image maps</i>	9
508 (f) <i>Client-side image maps</i>	9
508 (g) <i>Table headers</i>	10
508 (h) <i>Data Tables</i>	10
508 (i) <i>Frames</i>	12
508 (j) <i>Blinking Text</i>	13
508 (l) <i>Plug-ins</i>	14
508 (m) <i>Links to Plug-ins</i>	14
508 (n) <i>Forms</i>	14
508 (o) <i>Skip navigation</i>	16
508 (p) <i>Timed response</i>	17
The Web Content Accessibility Guidelines (WCAG)	17
<i>W3C WAI recommends using WCAG 2.0, instead of WCAG 1.0</i>	17
WCAG Checkpoints not in 508 for Priority 1 Compliance	18
<i>WCAG 1.3 Auditory description</i>	18
<i>WCAG 4.1 Natural Language</i>	18
<i>WCAG 6.2 Dynamic Content</i>	18
<i>WCAG 6.3 Scripting</i>	19
<i>WCAG 14.1 Auditory description</i>	19
Tools for Developing and Evaluating Accessible Content	19
Creating Accessible Documents	20
<i>Authoring Accessible MS Office Documents</i>	20
<i>Quick Tips to Accessible Word Processing Documents</i>	20
<i>PDF Documents Types</i>	20
<i>Generating PDF documents</i>	21

Introduction

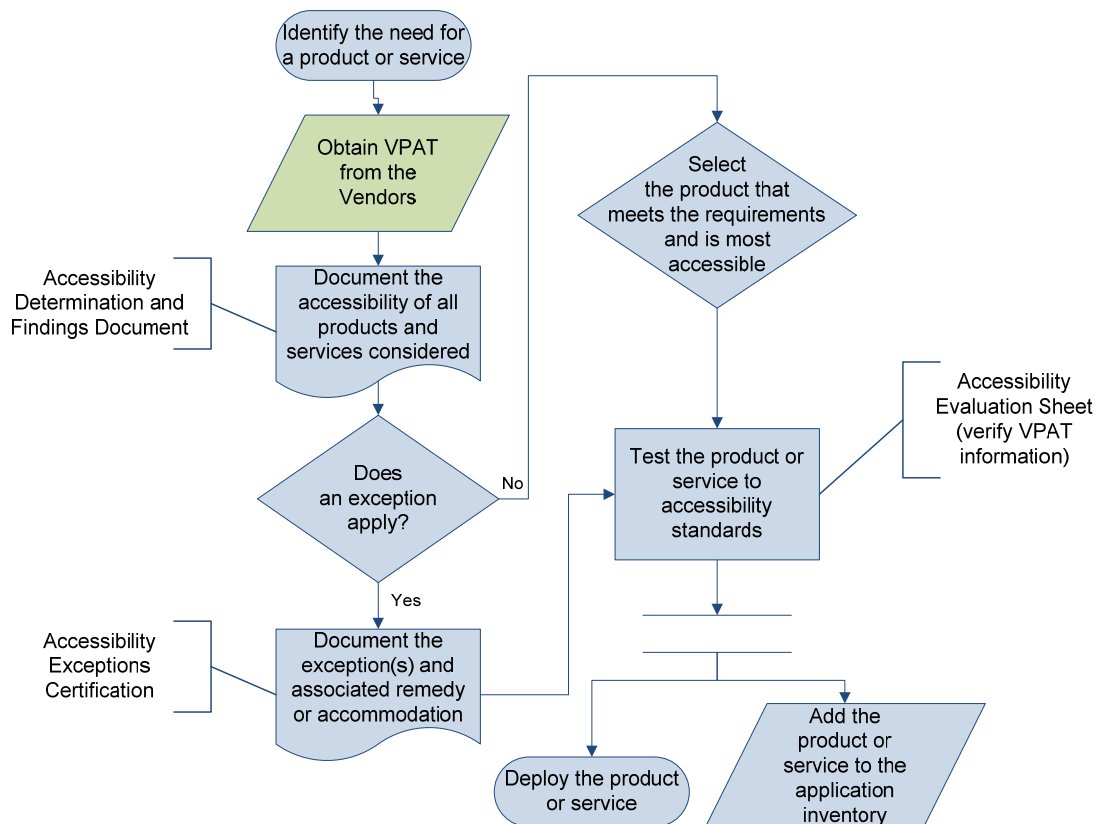
In April 2008, the United States Access Board published a report that contains a set of recommended standards and guidelines that the Access Board may use to update regulations that implement two laws regarding accessible information and communication technology (ICT): Section 508 of the Rehabilitation Act and Section 255 of the Communications Act of 1996. These two laws help to form the legal backbone of accessibility in the American ICT environment.

One of the first approved steps taken to make content available to disadvantaged users, was to use text equivalent pages as developed by Usablenet. While this can often provide an accessible way to view an inaccessible site, it should never be looked at as an alternative to actually making the original site accessible. Text only sites are an outdated concept for a number of reasons, and if we make the primary sites accessible, you can offer your users a variety of views based on their needs using CSS with minimal effort. In addition, because of expected future changes in these federal laws regarding accessibility, all new projects should work with the PMO office to ensure development effort will conform to standards expected to be in place at implementation

Though current law directs development conform to WCAG 1.0, development using WCAG 2.0 should result in not only a greater degree of accessibility, but should require for less rework when new Section 508 standards are ultimately published.

Accessibility in Procurement

The following diagram illustrates the process that should be followed ensure accessibility adequate consideration of EIT in the procurement process.



Market Research and Findings

Requesting officials must perform market research on desired business needed products. If during market research, it is determined that any portion of the requirement meets the definition of electronic and information technology (EIT), the program official should:

- Complete the Accessibility Determination and Findings Document to record their findings.

If an exemption applies, identify which exemption applies by:

- Checking it off and answer the appropriate questions.

The requisition creator and approving official are responsible for determining if a vendor's product complies with the Section 508 Standards. Based on the results of the Determination and Findings Document the requisition creator may be required to complete a Accessibility Exceptions Certification.

If there is an exception, there may be no need for accessibility testing, otherwise the product of service should be tested to determine the accessibility in accordance with the accessibility standards.

Finally, upon deployment the application inventory for the System Office or College should be updated to include the addition of a product or service including information about the extent of accessibility of the product or service.

Please forward your market research along with documents to your servicing procurement or purchasing office and a copy to the IT Accessibility Coordinator, VCCS IT Systems Office; or Fax: 804-819-4774. For further information visit: <http://www.vccs.edu/FacultyStaff/InformationTechnology/ITAccessibility> or for tools to help you in your research you can visit GSA's Buy Accessible Wizard at: <http://applbuyaccessible.gov/baw>

Accessible Evaluation Sheet

For EIT, covered under Section 508 subpart B, that is developed or is a combination of (COTS) Commercial Off-the-Shelf Products, you should attach an Accessibility Evaluation Sheet that is to be filled out by the Requiring Official and validated by an Information Technology specialist prior to the purchase of the product and/or service.

This guide is for use of buyers, and developers only and helps them to evaluate various products and services. The commercial availability is determined using the Market Research. This is recorded in the column headed 'Commercial Availability'. Provisions determined to be commercially available in pre-solicitation market research for all products / services reviewed are indicated by the phrase 'partially or fully available in all'. Provisions determined to be commercially available in pre-solicitation market research for some products / services reviewed are indicated by the phrase 'partially or fully available in some'. Provisions not determined to be commercially available in pre-solicitation market research have no value in the Commercial Availability column.

What and Why of Accessible Web Design

The Web is an important medium for receiving information as well as for providing information and interacting with society. Therefore, it is essential that the Web is accessible in order to provide equal access and equal opportunity to people with disabilities. This basic human right is recognized in the UN [Convention on the Rights of Persons with Disabilities](#), which specifically mentions the Internet and other accessible information and communications technologies (ICT). An accessible Web can also help people with disabilities and older people more actively participate in society.

Accessible Web design is the process of ensuring that users are able to use a Web resource regardless of abilities and disabilities without further modification.

Making A Web Site Accessible Both For People With Disabilities And For Mobile Devices

With global mobile phone use at an all time high, there has been a surge of interest in developing Web sites that are accessible from a mobile device. Similarly, making Web sites accessible for people with disabilities is an integral part of high quality Web sites, and in some cases a legal requirement.

Most Mobile Web specialists don't know about design issues for people with disabilities. Likewise, most Web accessibility specialists don't know Mobile Web design best practices.

Web sites can more efficiently meet both goals when developers understand the significant overlap between making a Web site accessible for a mobile device and for people with disabilities. The similarities are introduced below along with benefits of addressing both and resources with technical details of the overlap.

Similar Barriers

Users of mobile devices and people with disabilities experience similar barriers when interacting with Web content. For example, mobile phone users will have a hard time if a Web site's navigation requires the use of a mouse because they typically only have an alphanumeric keypad. Similarly, desktop computer users with a motor disability will have a hard time using a Web site if they can't use a mouse. Additionally, people with disabilities may use a mobile device to access the Web site.

Similar Solutions

The W3C provides guidelines/standards on making accessible Web content and best practices for content mobile-friendly.

The Web Content Accessibility Guidelines (WCAG) is a guide for making Web sites accessible to people with disabilities, and the Mobile Web Best Practices (MWBP) is a guide for making Web sites usable from a mobile device.

There is an overlap between MWBP and WCAG. For example, the MWBP best practice "Label all form controls appropriately and explicitly associate labels with form controls" corresponds with the WCAG 1.0 checkpoint "12.4 Associate labels explicitly with their controls" and WCAG 2.0 technique "H44: Using label elements to associate text labels with form controls".

Doing Both

Following these two guidelines makes your Web content more accessible to everyone regardless of situation, environment, or device. Designing to the guidelines together, instead of separately, can make the process more efficient.

Web sites that already meet WCAG or MWBP are already well on the way to meeting the other.

Understanding the overlap also strengthens the business case for adopting WCAG or MWBP in a Web site that already complies with one, or for adopting both together.

Section 508

"Section 508" refers specifically to Section 508 of the Rehabilitation Act of 1973, as amended by the Workforce Investment Act of 1998. The law requires Federal agencies to purchase electronic and information technology that is accessible to employees with disabilities, and to the extent that those agencies provide information technology to the public, it too shall be accessible by persons with disabilities.

Actually Section 508 was included in an amendment to the Rehabilitation Act in 1986, with the requirement that the Federal Government provide accessible technology to employees and to the public. But the 1986 version provided no guidance for determining accessibility of information technology and there were no enforcement procedures.

The 1998 amendment addressed both these issues. The Access Board (the Architectural and Transportation Barriers Compliance Board) was assigned the task of determining standards for accessible electronic and information technology, although the law applies to the development, procurement, maintenance, or use of all electronic and information technology.

Section 508 Described

508 (a) Text Equivalent

A text equivalent for every non-text element shall be provided (e.g., via "alt", "longdesc", or in element content (similar to WCAG 1.1).

Explanation:

Every graphic must have text description to explain what the graphic shows. The general rule when using images on a web page is to provide an alt attribute in the image code and to provide the same information which the visual user sees. This is called "providing an equivalent experience." An alt tag should be meaningful and contain no more than 9 words.

When a graphic includes text as part of the graphic (in element content), provide an alt attribute that includes the depicted text.

When describing simple graphics or photos, use the `<alt=" ">` tag in your `` field. Animated graphics should be described, as well, to indicate the action (see example below).

Bullets, horizontal rules, transparent images used as placeholders, or images which have no significant meaning for the page (what some webmasters call "eye candy") can be coded as `<alt="">`

Many web editors provide a field for inserting the alt tag. In MS FrontPage, right-click on your graphic and select Image Properties; in MS Word right-click on your graphic, select size then the Alt Text tab, there you will find the alt text field for inserting your text.

Examples:

"I contemplate the University of Virginia as the future bulwark of the human mind in this hemisphere."

—THOMAS JEFFERSON, 1820



Image source code:

```
<img src ="uvalogo.gif" alt="The University of Virginia Rotunda">
```

NOTE: If you have a logo which represents your unit on the page, it is not enough to just say alt="UVA logo." Such a tag would "pass" various accessibility tests, but obviously this tag is not providing an "equivalent experience." A better alt attribute would be: alt="The University of Virginia Rotunda." This is the text which conveys the information shown on the logo.



In element content:

Image source code:

```
<img src ="/portals/0/skins/vccs/images/vccs_logo_large.gif" alt="The Virginia Community College System logo and the URL:myfuture.vcss.edu under the logo">
```

Animated image:

Animated image source code:

```
<img src ="pubsbook.gif" alt="animated book flips through pages">
```

Bullet:



Image source code:

```
<img src ="roudbullet.gif " alt="*">
```

Long description file

In order to facilitate screen readers reading a page, one way to describe a complex chart or graphic in more detail is to use a description link, normally called a D-link. Rather than spell out "description link," the convention is to use a single letter of D near the chart, which (when selected) takes the user to another document describing the chart/graphic. The D can be very small, such as 2 or 4 point or may be the same color as the background.

```
<img src ="enrollment.gif" alt="2010 undergraduate enrollment"><a href=longdesc.htm">D</a>
```

In this coding (above), a sighted person would see the D next to the chart, and a screen reader would speak the letter D.

You can also make the D invisible, still placed next to the image, but having an alt text of D. The invisible D might be a transparent gif with D as the alt text, or the letter D coded to the same color as the web document background.

```
<img src ="enrollment.gif" alt="2001 undergraduate enrollment"><a href=longdesc.htm"></a>
```

Make the D into a link to a file named "longdesc.htm" or "longdesc.html." In this longdesc file, you can write a full description of the graphic. If you have more than one photo being described in the longdesc file, use a bookmark to take the user directly to the description of the related photograph.

508 (b) Equivalent Multimedia Alternatives

Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation (same as WCAG 1.4).

Explanation:

Providing equivalent information for inaccessible content is one of the primary ways authors can make their documents accessible to people with disabilities.

What is meant by equivalent?

- For graphic and audio information this would be a text equivalent, which should convey all essential content. Text content can be presented to the user as synthesized speech, Braille, and visually-displayed text.
- Non-text equivalents (e.g., an auditory description of a visual presentation, a video of a person telling a story using sign language as an equivalent for a written story, etc.) also improve accessibility for people who cannot access visual information or written text, including many individuals with blindness, cognitive disabilities, learning disabilities, and deafness.

Equivalent information may be provided in a number of ways, including through attributes (e.g., a text value for the "alt" attribute in HTML and SMIL), as part of element content (e.g., the OBJECT in HTML), as part of the document's prose, or via a linked document (e.g., designated by the "longdesc" attribute in HTML or a description link). Depending on the complexity of the equivalent, it may be necessary to combine techniques (e.g., use "alt" for an abbreviated equivalent, useful to familiar

readers, in addition to "longdesc" for a link to more complete information, useful to first-time readers).

A text transcript is a text equivalent of audio information that includes spoken words and non-spoken sounds such as sound effects. A caption is a text transcript for the audio track of a video presentation that is synchronized with the video and audio tracks. Captions are generally rendered visually by being superimposed over the video, which benefits people who are deaf and hard-of-hearing, and anyone who cannot hear the audio (e.g., when in a crowded room)

For use on Windows based platforms, the Media Access Generator, MAGpie, is an authoring tool for making video and multimedia materials accessible to persons with disabilities. It provides an environment for authors who want to add captions, subtitles and audio descriptions to their work. It's available from the CPB/WGBH National Center for Accessible Media (NCAM):
<http://ncam.wgbh.org/webaccess/magpie/>

Examples:

A text transcript is a text equivalent of audio information that includes spoken words and non-spoken sounds such as sound effects.

A caption is a text transcript for the audio track of a video presentation that is synchronized with the video and audio tracks. Captions are generally rendered visually by being superimposed over the video, which benefits people who are deaf and hard-of-hearing, and anyone who cannot hear the audio (e.g., when in a crowded room).

A collated text transcript combines (collates) captions with text descriptions of video information (descriptions of the actions, body language, graphics, and scene changes of the video track). These text equivalents make presentations accessible to people who are deaf-blind and to people who cannot play movies, animations, etc. It also makes the information available to search engines.

One example of a non-text equivalent is an auditory description of the key visual elements of a presentation. The description is either a prerecorded human voice or a synthesized voice (recorded or generated on the fly). The auditory description is synchronized with the audio track of the presentation, usually during natural pauses in the audio track. Auditory descriptions include information about actions, body language, graphics, and scene changes.

See National Center for Accessible Media (<http://ncam.wgbh.org/richmedia/examples/index.php>) for examples of equivalent alternatives for any multimedia presentation.

508 (c) Color

Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup (same as WCAG 2.1).

Explanation:

Because some users cannot distinguish colors or shades of colors, do not use color to exclusively define the navigation of your site. For example, do not tell the user to "Follow the blue buttons to learn about Student Life, and the purple buttons to learn about Class Signup Procedures." Color may be used, but in conjunction with text (on the buttons or next to the buttons). Therefore, you might make the Student Life buttons blue, but you would have the appropriate text on or next to them to explain where the button will take the user.



Student Life

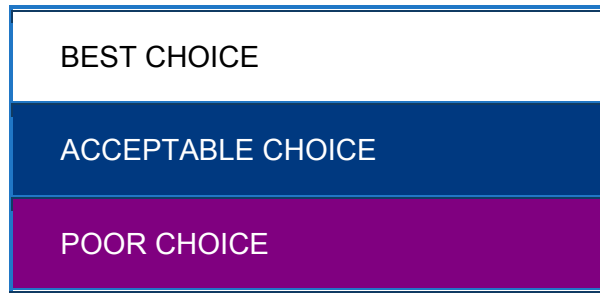


Class Signup
Procedures

High contrast

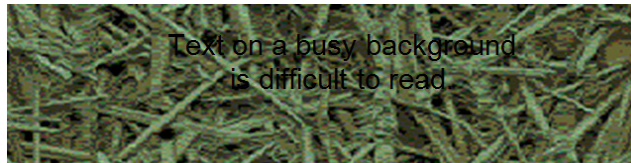
High contrast between elements and between the background and text is essential for most users. For example, most users will have a difficult time distinguishing between a purple background and blue

text. Very light or white backgrounds with dark or black text work well, as do very dark or black backgrounds with white or very light text. Large amounts of light text on a dark background is tiring to read and will slow most readers down, so it is best to set body copy in a dark color on a light background.



Graphical backgrounds

Avoid using busy, complicated background pictures on web pages. Brickwork or wickerweave, for example, might add interest, but reading anything other than large text against such a background is difficult.



508 (d) Style sheet use

Documents shall be organized so they are readable without requiring an associated style sheet (same as WCAG 6.1).

Explanation:

Style sheets are often used for controlling the look and layout of a page. They may specify fonts, color, indentation and other information. However, the page should still be understandable (even if less visually appealing) when the style sheet is turned off. If the document is confusing or information is missing when the style sheet is turned off, the page should be redesigned.

508 (e) Server-side image maps

Redundant text links shall be provided for each active region of a server-side image map (same as WCAG 1.2)

Explanation:

An image is a server-side image map if the bottom of your browser screen indicates the X, Y coordinates of your mouse while you move it over the image.

Server-side image maps do not send imbedded links to the user of the image map. Therefore, the user will not be able to access the links in the image map. You must provide text links adjacent to the server-side image map, which provide the same links as the image map.

508 (f) Client-side image maps

Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape (same as WCAG 9.1).

Explanation:

Client-side image maps have been a feature of HTML since 1994. Use your web editor to create hotspots in your image map and to insert the alt tags. An image map saved to your server is a client-

side image map.

508 (g) Table headers

Row and column headers shall be identified for data tables (same as WCAG 5.1).

Explanation:

If you provide data in a table, place the column or row labels within the first row or column of the table. The reason for this: When they are placed outside the table, the labels cannot be associated with the specific row or column they are meant to identify.

Examples:

Item	Price
Textbook A	\$39.95
Textbook B	21.65

In this example, Item and Price identify data in the columns below them. The user of a screen reader can now choose to read down a column, knowing what the column contains.

To identify a header cell, use following markup: `<th>data</th>`. FrontPage allows you to define a header cell in the Cell Properties dialog box.

The following example shows the wrong way to create this table. The column headers are no longer part of the column, and the user of a screen reader would not be able to connect the column headers with the column contents.

Item	Price
Textbook A	\$39.95
Textbook B	21.65

508 (h) Data Tables

Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers (same as WCAG 5.2)

Explanation:

To group table rows in HTML, use THEAD for repeated table headers, TFOOT for repeated table footers, and TBODY for other groups of rows. Use COL and COLGROUP to group columns, and the "axis", "scope", and "headers" attributes, to describe more complex relationships among data.

Example:

(From the W3 guidelines: <http://www.w3.org/TR/WCAG10-HTML-TECHS/#identifying-table-rows-columns>)

This table lists travel expenses at two locations: San Jose and Seattle, by date, and category (meals, hotels, and transport).

```
<table border="1">
<caption>Travel Expense Report</caption>
<tr>
<th></th>
<th id="header2" axis="expenses">Meals
<th id="header3" axis="expenses">Hotels
<th id="header4" axis="expenses">Transport
<td>subtotals</td>
```

```
<tr>
<th id="header6" axis="location">San Jose
<th> <th> <th> <td>
<tr>
<td id="header7" axis="date">25-Aug-97
<td headers="header6 header7 header2">37.74
<td headers="header6 header7 header3">112.00
<td headers="header6 header7 header4">45.00
<td>
<tr>
<td id="header8" axis="date">26-Aug-97
<td headers="header6 header8 header2">27.28
<td headers="header6 header8 header3">112.00
<td headers="header6 header8 header4">45.00
<td>
<tr>
<td>subtotals
<td>65.02
<td>224.00
<td>90.00
<td>379.02
<tr>
<th id="header10" axis="location">Seattle
<th> <th> <th> <td>
<tr>
<td id="header11" axis="date">27-Aug-97
<td headers="header10 header11 header2">96.25
<td headers="header10 header11 header3">109.00
<td headers="header10 header11 header4">36.00
<td>
<tr>
<td id="header12" axis="date">28-Aug-97
<td headers="header10 header12 header2">35.00
<td headers="header10 header12 header3">109.00
<td headers="header10 header12 header4">36.00
<td>
<tr>
<td>subtotals
<td>131.25
<td>218.00
<td>72.00
<td>421.25
<tr>
<th>Totals
<td>196.27
<td>442.00
```

```
<td>162.00
<td>800.27
</table>
```

Here is how it might look in a browser:

Travel Expense Report

	Meals	Hotels	Transport	subtotals
San Jose				
25-Aug-97	37.74	112.00	45.00	
26-Aug-97	27.28	112.00	45.00	
subtotals	65.02	224.00	90.00	379.02
Seattle				
27-Aug-97	96.25	109.00	36.00	
28-Aug-97	35.00	109.00	36.00	
subtotals	131.25	218.00	72.00	421.25
Totals	196.27	442.00	162.00	800.27

508 (i) Frames

Frames shall be titled with text that facilitates frame identification and navigation (same as WCAG 12.1).

Explanation:

Frames are often used to organize a page into logical units of information. The relationship between the content in the various frames (e.g., one frame has a table of contents; another, the contents themselves) is readily apparent to the sighted user; however, those relationships must be conveyed by some other means for non-visual users.

There are a number of reasons why frames present challenges to web accessibility:

- Without scripting, they tend to break the "previous page" functionality offered by browsers.
- It is impossible to refer to the "current state" of a frameset with a URI (Uniform Resource Identifier, the preferred term used by the World Wide Web Consortium for a "web address"); once a frameset changes contents, the original URI no longer applies.
- Opening a frame in a new browser window can disorient or simply annoy users. (However, opening another site's page within your framed page can imply you are the document creator; increasingly this is a touchy (legal) issue.)

Put a title on each frame

```
<HTML>
<HEAD>
<TITLE>A simple frameset document</TITLE>
</HEAD>
<FRAMESET cols="10%, 90%"
title="Our library of electronic documents">
<FRAME src="nav.html" title="Navigation bar">
<FRAME src="doc.html" title="Documents">
</FRAMESET>
```

Include a NOFRAMES element

For every occurrence of FRAMESET, include immediately after it a <NOFRAME> element that directs a user to the appropriate web page. You may also want to describe the layout and purpose of frames and how multiple frames relate to each other. In the example below, this is called frameset-desc.html.

```
<FRAMESET cols="10%, 90%"
title="Our library of electronic documents">
<FRAME src="nav.html" title="Navigation bar">
<FRAME src="doc.html" title="Documents">
<NOFRAMES>
<A href="noframes.html" title="Library link"> Select to go to the electronic library</A>
<p><a href="frameset-desc.html">Descriptions of frames.</a></p>
</NOFRAMES>
</FRAMESET>
```

The frameset-desc.html might say something like:

nav.html - this frame provides common navigation links for the site
doc.html - this frame lists documents available electronically

508 (j) Blinking Text

Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz (similar to WCAG 7.1).

Explanation:

Avoid using the BLINK and MARQUEE elements. Use another method, such as text size, to draw attention to the text. Text coded to BLINK or MARQUEE is not readable by screen readers. Also text displayed this way can be difficult to understand by users with learning disabilities. Furthermore, as is stated in the World Wide Web Consortium guidelines on accessibility, people with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 2 to 55 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).

508 (k) Dynamic HTML or JavaScript

A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes (same as WCAG 11.4).

Explanation:

Many elements of Dynamic HTML (client-side scripted HTML, which is usually accomplished with JavaScript) cannot be made directly accessible to assistive technologies and keyboards, especially when the onMouseover command is used. If an onMouseover (or similar) element does not contain any important information (e.g. the script causes a button to "glow"), then there is no consequence for accessibility. If this scripted event reveals important information, then a keyboard-accessible alternative is required.

Text equivalents of frames should be provided so that their contents and the relationships between frames make sense. This also means that when the contents of a frame changes, the description, if any, must change too. You can't do that if an IMG is inserted directly into a frame. Therefore, the source ("src") of a frame should always be an HTML file, and any images, along with their text alternatives, should be placed in the HTML file.

Example:

A text-only version of the page should be used only when there is no other way to make the content accessible, or when it offers significant advantages over the "main" version for certain disability types. It must be kept up-to-date with the primary version.

When components (e.g. plug-ins, scripts) are not directly accessible, an alternative needs to be provided

Scripts (e.g. JavaScript pop-up menus) should either be directly accessible to assistive technologies (keyboard accessibility is a good measure of this), or an alternative method of accessing equivalent functionality should be provided (e.g. a standard HTML link).

508 (l) Plug-ins

When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology (Similar to WCAG 6.3 however WCAG 6.3 is more restrictive (see page 19, WCAG 6.3 Scripting)).

Explanation:

Programmatic objects, such as scripts, applets, and other plug-ins are not HTML and have their own sets of access issues. It is important to attend to these issues in order to ensure that the web page as a whole is accessible. They can cause a number of different accessibility barriers:

Blinking text and motion that is gratuitous or that the user cannot stop can be difficult to read and/or distracting.

A refresh or redirect of a page may cause the content to change unexpectedly, when the user is not prepared for it. For example, the user may be sent to a new page while a screen reader is in the middle of reading a sentence.

Examples:

```
<applet name="DigiChat"
"codebase="http://rm150nt.cpd.usu.edu/DigiChat/DigiClasses/"
code="digi/digichat/DigiChatApplet.class"
width="200" height="100" align="right"
archive="Client.jar"
alt="This is a chat program which requires a Java-compatible
Web Browser to run">
<param name="port" value="8303">
<param name="background" value="FFFFFF">
<param name="textcolor" value="000000">
<param name="cabbase" value="Client.cab">
</applet>
```

508 (m) Links to Plug-ins

When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with §1194.21(a) through (l); (Similar to WCAG 6.3).

Examples:

Provide a link to Adobe Acrobat Reader when using PDF files.

```
<font face="Arial">Provide a link to <a
href="http://www.adobe.com/products/acrobat/readstep2.html">Adobe Acrobat Reader</a> when using
PDF files.</font>
```

508 (n) Forms

When electronic forms are designed to be completed online, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues (Similar to WCAG 10.2)

Explanation:

In order for a person using a screen reader to successfully navigate a form, each of the form's labels must be linked to its controls by HTML markup.

To create accessible forms:

1. Place form labels adjacent to their corresponding form controls.
2. Use HTML markup to associate controls explicitly with their labels. For example:
3. Provide markup for labels, using the `<label>` tag.
4. Group related form elements using the `<fieldset>` tag.
5. Group checkboxes in a fieldset tag, and provide a `<label>` for each checkbox.
6. Provide a title or "legend" for each fieldset using the `<legend>` tag.
7. Always provide a button for users to submit forms. Don't use JavaScript to automatically submit forms.

Example:

First Name	<input type="text"/>								
Last Name	<input type="text"/>								
Phone	<input type="text"/>								
Email	<input type="text"/>								
<table> <tr> <td>Class of</td> <td>Gender</td> </tr> <tr> <td><input checked="" type="radio"/> 2004</td> <td><input checked="" type="radio"/> Male</td> </tr> <tr> <td><input type="radio"/> 2005</td> <td><input type="radio"/> Female</td> </tr> <tr> <td><input type="radio"/> 2006</td> <td></td> </tr> </table>		Class of	Gender	<input checked="" type="radio"/> 2004	<input checked="" type="radio"/> Male	<input type="radio"/> 2005	<input type="radio"/> Female	<input type="radio"/> 2006	
Class of	Gender								
<input checked="" type="radio"/> 2004	<input checked="" type="radio"/> Male								
<input type="radio"/> 2005	<input type="radio"/> Female								
<input type="radio"/> 2006									

Source code:

```
<form name="form2" action method="post">
<table cellSpacing="0" cellPadding="4" width="50%" border="0" style="border-collapse: collapse"
bordercolor="#111111">
<td>
<div align="right">
<label for="fname1">First Name</label></div>
</td>
<td><input id="fname1" size="12" name="textfield33"> </td>
</tr>

<tr>
<td>
<div align="right">
<label for="lname1">Last Name</label></div>
</td>
<td><input id="lname1" size="12" name="textfield34"> </td>
</tr>

<tr>
<td>
<div align="right">
<label for="phone1">Phone</label></div>
</td>
<td><input id="phone1" size="12" name="textfield35"> </td>
```

```

</tr>

<tr>
<td>
<div align="right">
<label for="email1">Email</label></div>
</td>
<td><input id="email1" size="12" name="textfield36"> </td>
</tr>

<tr>
<td bgColor="#CCF4FF" align="left" valign="top"><fieldset>
<legend>Class of</legend>
<br>
<input id="210" type="radio" value="radiobutton13" name="radiobutton3" checked>
<label for="210">2004</label><br>
<input id="211" type="radio" value="radiobutton14" name="radiobutton3">
<label for="211">2005</label> <br>
<input id="212" type="radio" value="radiobutton15" name="radiobutton3">
<label for="212">2006</label>
</fieldset></td>

<td bgColor="#ccccff" align="left" valign="top"><fieldset>
<legend>Gender </legend>
<br>
<input id="male1" type="radio" value="radiobutton9" name="radiobutton4" checked>
<label for="male1">Male</label> <br>
<input id="female1" type="radio" value="radiobutton10" name="radiobutton4">
<label for="female1">Female</label><br>
</fieldset></td>
</tr>

<tr>
<td colspan="2">
<p align="center"><input type="button" value="Submit" name="B1">
</td>
</tr>

</table>
</form>

```

508 (o) Skip navigation

A method shall be provided that permits users to skip repetitive navigation links, (Related to WCAG 13.5 and 13.6, however 508 is more strict).

Explanation:

Many sites have a uniform set of navigation links that appear on all or most of the pages. The repeated reading of these navigation links can become quite tedious for those using assistive technologies such as screen readers. The good news is that it's easy to provide a means to skip over such sections. Place an anchor (or bookmark) at the start of your main content. Then, place a hyperlink to that anchor before the start of your navigational links. And, for browsers which support the HTML 4.0 "tabindex" attribute, you should include "tabindex=1" in the anchor after the links. This permits users to tab past the redundant links.

Examples:

```
<a href="#main_content">Skip navigational links</a>
```

```
.....
```

```
<a name="main_content" tabindex=1></a>
```

508 (p) Timed response

When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.

Explanation:

When a timed response is required, as in an on-line quiz, the user shall be alerted and be given sufficient time to indicate that more time is required.

Example:

For more information, see: [http://www.access-board.gov/sec508/guide/1194.22.htm#\(p\)](http://www.access-board.gov/sec508/guide/1194.22.htm#(p))

The Web Content Accessibility Guidelines (WCAG)

The [Web Accessibility Initiative \(WAI\)](#) was formed by the [World Wide Web Consortium \(W3C\)](#) in order to bring accessibility considerations into the technology development of the Web Consortium and to determine guidelines for accessible technology including web authoring and user agents (browsers). As Tim Berners-Lee, the inventor of the Web, and the Director of the W3C put it, "The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect."

The first version of the authoring guidelines, the [Web Content Accessibility Guidelines 1.0](#), became a W3C Recommendation on May 5, 1999.

The guidelines are further organized into a [checklist](#). The checkpoints are categorized as Priority 1, 2 or 3. Here is the characterization of those priorities from the Guidelines:

[Priority 1]

A Web content developer must satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.

[Priority 2]

A Web content developer should satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.

[Priority 3]

A Web content developer may address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.

Web Content Accessibility Guidelines 1.0 was published in May 1999. WCAG 2.0 was published on 11 December 2008. WCAG 2.0 applies broadly to more advanced technologies; is easier to use and understand; and is more precisely testable with automated testing and human evaluation.

W3C WAI recommends using WCAG 2.0, instead of WCAG 1.0.

Most Web sites that conform to WCAG 1.0 will not require significant changes in order to conform to WCAG 2.0, and some may not need any changes. To help you move to WCAG 2.0, WAI is developing:

- [How WCAG 2.0 Differs from WCAG 1.0](#)

- [Comparison of WCAG 1.0 Checkpoints to WCAG 2.0](#)
- [How to Update Your Web Site from WCAG 1.0 to WCAG 2.0](#)

WCAG Checkpoints not in 508 for Priority 1 Compliance

Four WCAG Priority 1 checkpoints, 1.3, 4.1, 6.2 and 14.1, are listed as "not in 508" in the Comparison column of this table. If a web site is 508-compliant and its author wants to be Web Accessibility Initiative A-Compliant as well, there are the only four checkpoints that must be addressed additionally, checkpoints, 1.3, 4.1, 6.2 and 14.1.

WCAG 1.3 Auditory description

Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation Explanation:

Explanation:

By WCAG 1.1 and 1.4 (Section 508 1194.22 (a) and (b)) video must have a synchronized text equivalent. Given the web environment it is natural to assume that the synchronized text equivalent could be displayed in a window next to (or above or below) the video just like captions. The problem addressed by WCAG 1.3 is that blind users, for whom this is important, do not today have access to that text; their screen readers won't read the descriptions of the video. Until they do, WCAG 1.3 requires that the text description of the video be presented in audio.

Video on the web which has text descriptions of important video information will conform to the Section 508 web standards.

However, in the discussion of the standards, the Access Board specifically referred to the multi-media section of the standards:

The Board did not adopt WCAG 1.0 Checkpoint 1.3 which provides that "[u]ntil user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation...." Although the NPRM did not propose addressing this issue in the web section, there was a similar provision in the multi-media section of the NPRM.

Indeed there is a similar provision in the final rule as well. Paragraph 1194.24 (d) of the multi-media section (cited above) requires that training and informational multi-media productions which support the agency's mission shall have audio descriptions.

WCAG 4.1 Natural Language

Clearly identify changes in the natural language of a document's text and any text equivalents (e.g., captions).

Explanation:

The Access Board determined that:

The intent of 4.1 is to for web authors to indicate change in natural language with markup (`lang="en"`), not using in-line text, like "the following is in German."

Not many assistive technologies support language change markup.

Based on that determination, the Access Board decided not to include this checkpoint as a standard for Section 508.

WCAG 6.2 Dynamic Content

Ensure that equivalents for dynamic content are updated when the dynamic content changes.

Explanation:

The Access board did not include this checkpoint in the Section 508 standards for web accessibility because it was deemed unclear.

The purpose of Checkpoint 6.2 is to back up other checkpoints, like 6.3, that require text alternatives for dynamic content. Checkpoint 6.2 says the text alternatives must be kept up-to-date. The techniques document for this checkpoint (<http://www.w3.org/TR/WCAG10-HTML-TECHS/#scripts-alt>) gives an example of using the NOSCRIPT element displaying sports scores in a definition list while the script would present the scores in a "bill board." This checkpoint requires that these two presentations are displaying the same scores.

Another example of this, my favorite, is a JavaScript function that displays the date the page was last updated at the bottom of a web page by querying the file date. This can ensure that the update information is current without having to change the update information every time the page is modified. But if you use the NOSCRIPT option as a text alternative to that dynamic content, the NOSCRIPT content would have to be updated every time the page was modified by this checkpoint, thereby nullifying the usefulness of the script.

WCAG 6.3 Scripting

Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.

Explanation:

The WCAG checkpoint is much easier to interpret; your pages have to be usable when scripts, applets and other programmatic objects are turned off. If your page satisfies this checkpoint then it is likely that you also satisfy the corresponding Section 508 standards cited above.

However, the presumption of the Section 508 standards is that scripting, applets and other programmatic objects will be turned on (and supported) and those all must be accessible. So, if your site uses scripting just for visual enhancements, like changing text attributes when the mouse moves over text, then the site satisfies both WCAG 6.3 and Paragraph 1194.22 (l).

If you use "fly-over" menus implemented in JavaScript, and all the submenu items are available as normal text links, then the site satisfies both 6.3 and 1194.22 (l).

However, if you use Document.write to place (important) text on your page while it is loading, then it will be functional text available to assistive technology. Assuming that the text is important, the site fails WCAG 6.3 but passes 1194.22 (l).

WCAG 14.1 Auditory description

Use the clearest and simplest language appropriate for a site's content.

Explanation:

The Access Board decided against including this checkpoint as a standard for web accessibility because it was deemed too difficult to enforce. The requirement to use clearest and simplest language can be very subjective.

NOTE: If a web site is WCAG A-Compliant and its author wants to be Section 508 compliant as well, these are the five standards he must address additionally. These are paragraphs 1194.22 (l), (m), (n), (o), and (p).

Tools for Developing and Evaluating Accessible Content

Prior to deployment it is highly recommended that developers validate their efforts and record the results on a form such as the VCCS Accessibility Checklist.

Developing Accessible Applications, Microsoft <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vsent7/html/vxconDesigningAccessibleApplications.asp>

Java Accessibility Checklist, SUN... <http://www-03.ibm.com/able/guidelines/java/accessjava.html>

Web Accessibility Evaluation Tools <http://www.w3.org/WAI/ER/tools/> Prior to deployment developers should validate their

Creating Accessible Documents

Authoring Accessible MS Office Documents

Microsoft Office documents such as PowerPoint and Word are generally accessible to assistive technology like screen readers and voice recognition software. Keyboard equivalents are available for most functional activities.

Pictures inserted in documents should have alternate text added to for accessibility. Methods for adding the text varies based on the version.

Like most word processor, MS Word has built-in features that encode semantic structure and styles to control the visual presentation of structural elements. When converted, structural and styles elements translate into equivalent RTF, HTML, XML or PDF tags.

A well crafted MS Word document will convert easily into other file formats while maintaining all of its presentation and structural information

Quick Tips to Accessible Word Processing Documents

- When done correctly, the document ensures correct reading-order and allows it to reflow to fit the display when the magnification or display size is changed.
- Place content in logical reading order. This is fundamental to creating accessible documents.
- Do not use TAB, Spacebar and Enter keys to format for tables, columns, lists etc.
- Use the application's built-in features to encode semantic structure (headings, paragraphs, lists, sections, headers/footers, tables, columns, forms etc.).
- Define and use styles to format structural elements like headings, paragraphs etc. to control typography and layout.
- Use standard fonts. Do not use fonts that do not map to Unicode.
- Avoid complex layout, sidebars and other ornamentation as they make it difficult to maintain a logical reading order.
- Avoid placing content in drawing-canvases or text-boxes as these are floating objects and flow to the bottom of a page's reading-order.
- Group multiple graphic elements (created by drawing tools, charts etc) into one image.
- Provide alternative text descriptors for all non-textual elements (graphs, images, illustration, pictures, multimedia, etc) that provide essential information.
- Ensure that all navigation and interactivity can be performed using the keyboard.

PDF Documents Types

Image (unstructured) or a graphical representation of the original document. Since they are images, they are inaccessible to assistive technology like screen readers.

Searchable Image (structured) consisting of a combination of images (vector and raster) plus text from the original document. The text is searchable and thus is partially accessible to screen reader. These documents maintain the logical structure of the original document.

Tagged Document which is a true electronic document with semantic markup, presentation attributes, searchable text and information on non textual elements.

Similar to other mark-up languages, tagged PDF files have a hierarchy of "tags" encoding the semantic structure. The position of the tag within the tree represents the reading order of the document. When well crafted, these documents are accessible to assistive technology like screen readers. At this time only Acrobat Adobe is capable of creating a tagged PDF document.

Generating PDF documents

PDF documents are generally created:

- As a scanned image of a print document saved as PDF file (structured)
- From PostScript files or other publishing tools using Acrobat Distiller (structured)
- Directly "printed" from word processing, TeX or other publishing tools using other desktop and serverside PDF generators (structured /unstructured)
- "Printed" as a PDF file from any application using Mac OSX (unstructured)
- Generated from recent versions of MS Office (if images have alternate text added) and Adobe publishing tools (tagged)

The Systems office has a document, "Guide to Creating Accessible Documents" that is application specific to MS Word and Adobe Acrobat, and much more comprehensive than the following.

REFERENCES

Code of Virginia, § 2.2-2012	http://www.leg1.state.va.us/000/cod/2.2-2012.htm
Code of Virginia § 2.2-3500 et seq	http://www.leg1.state.va.us/000/cod/2.2-3500.htm
Code of Virginia § 51.5-1 et seq	http://leg1.state.va.us/cgi-bin/legp504.exe?000+cod+51.5-1
The United States Access Board	http://www.access-board.gov
Federal Government Site for Section 508 Information.....	www.section508.gov
Mobile Web Best Practices 1.0	http://www.w3.org/TR/mobile-bp/
VCCS ITS Accessibility Policy	http://www.vccs.edu/Portals/0/ContentAreas/ITS/Policies/VCCS-ITS-08-5100_Accessibility_Policy.pdf
VCCS Information Technology Accessibility Standard	http://www.vccs.edu/Portals/0/ContentAreas/ITS/Standards/ITS_Technology_Accessibility_Standard(2).pdf
Web Content Accessibility Guidelines (WCAG) 1.0	http://www.w3.org/TR/WCAG10/
Guide to Creating Accessible Documents.....	VCCS Systems Office

REVISION HISTORY

Date	Version	Author	Revision Description
08/26/2009	1.0	Abdulaziz Bulling	Original